

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №11
дисциплины «Программирование на Python»

Выполнил:
Данилецкий Дмитрий Витальевич
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А., канд. технических
наук, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

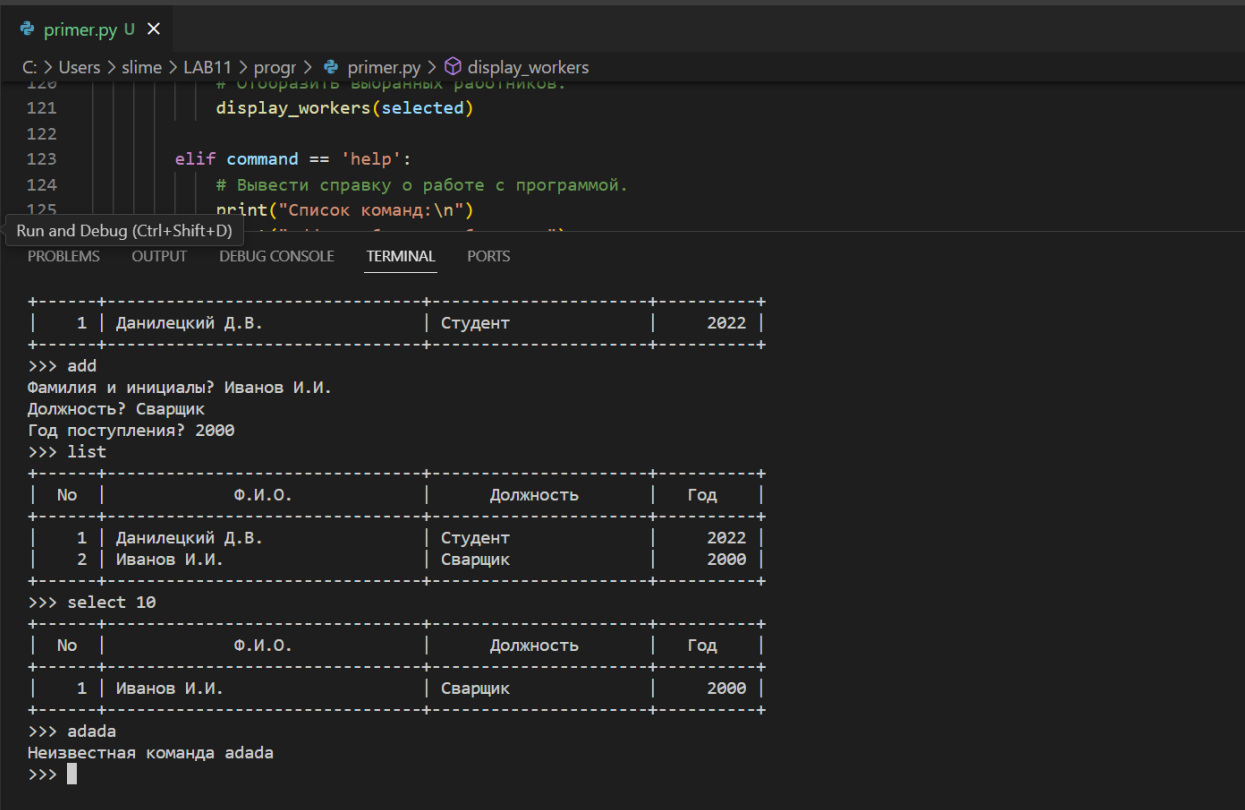
Ставрополь, 2023 г.

Тема: Работа с функциями в языке Python

Цель работы: приобретение навыков по работе с функциями при написании программ с помощью языка программирования Python версии 3.x.

Ход работы

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python. Выполнил клонирование созданного репозитория.
2. Дополнил файл .gitignore необходимыми правилами.
3. Организовал созданный репозиторий в соответствие с моделью ветвления git-flow.
4. Проработал пример лабораторной работы. Создал для него отдельный модуль языка Python. Привел в отчете скриншоты результата выполнения программы примера.

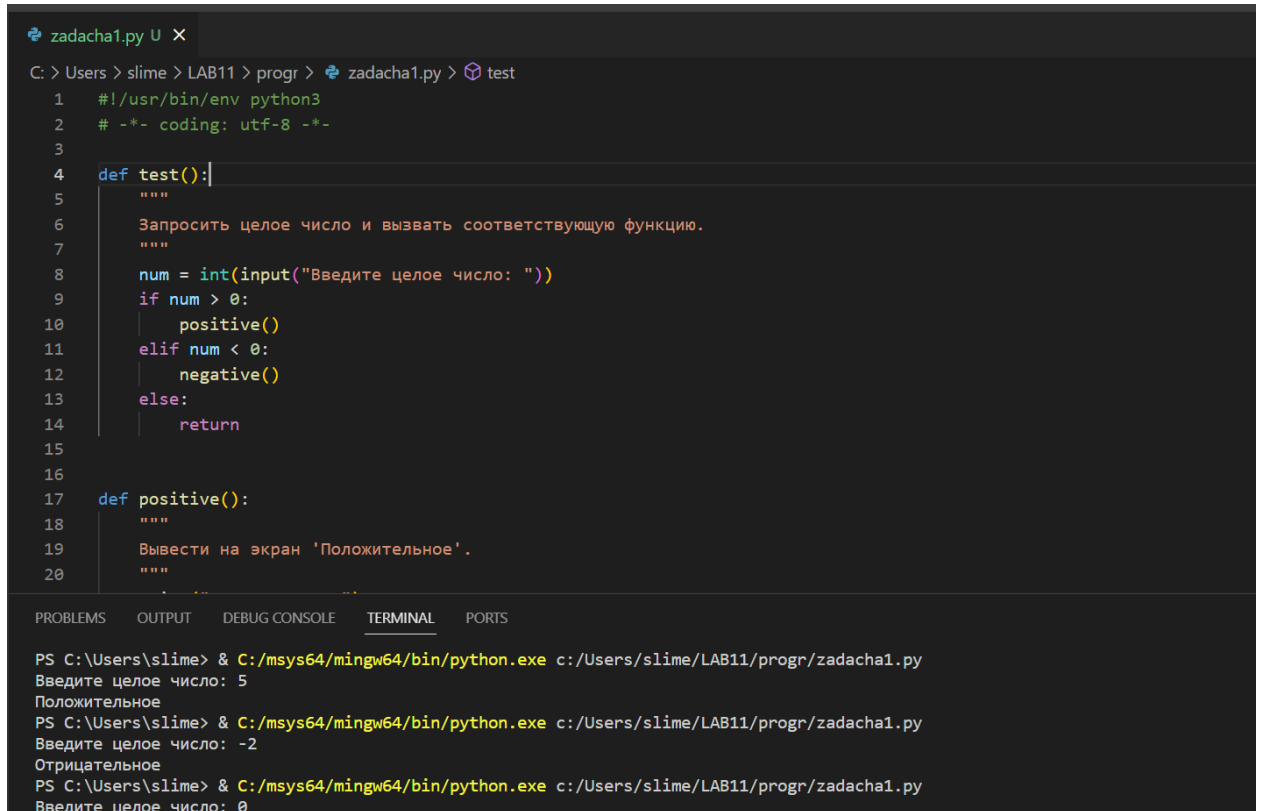


```
primer.py U X
C: > Users > slime > LAB11 > progr > primer.py > display_workers
120 # Отобразить выбранных работников.
121 display_workers(selected)
122
123 elif command == 'help':
124     # Вывести справку о работе с программой.
125     print("Список команд:\n")
Run and Debug (Ctrl+Shift+D)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
+-----+-----+-----+-----+
| 1 | Данилецкий Д.В. | Студент | 2022 |
+-----+-----+-----+-----+
>>> add
Фамилия и инициалы? Иванов И.И.
Должность? Сварщик
Год поступления? 2000
>>> list
+-----+-----+-----+-----+
| No | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Данилецкий Д.В. | Студент | 2022 |
| 2 | Иванов И.И. | Сварщик | 2000 |
+-----+-----+-----+-----+
>>> select 10
+-----+-----+-----+-----+
| No | Ф.И.О. | Должность | Год |
+-----+-----+-----+-----+
| 1 | Иванов И.И. | Сварщик | 2000 |
+-----+-----+-----+-----+
>>> adada
Неизвестная команда adada
>>> 
```

Рисунок 1. Результат работы программы из примера 1

5. Решил следующую задачу: основная ветка программы, не считая заголовков функций, состоит из двух строки кода. Это вызов функции test() и инструкции if __name__ == '__main__'. В ней запрашивается на ввод целое

число. Если оно положительное, то вызывается функция `positive()`, тело которой содержит команду вывода на экран слова "Положительное". Если число отрицательное, то вызывается функция `negative()`, ее тело содержит выражение вывода на экран слова "Отрицательное".



The screenshot shows a code editor window titled 'zadacha1.py' with the following Python code:

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def test():
5      """
6      Запросить целое число и вызвать соответствующую функцию.
7      """
8      num = int(input("Введите целое число: "))
9      if num > 0:
10         positive()
11     elif num < 0:
12         negative()
13     else:
14         return
15
16
17 def positive():
18     """
19     Вывести на экран 'Положительное'.
20     """
```

Below the code editor is a terminal window showing the execution of the program:

```
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slme/LAB11/progr/zadacha1.py
Введите целое число: 5
Положительное
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slme/LAB11/progr/zadacha1.py
Введите целое число: -2
Отрицательное
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slme/LAB11/progr/zadacha1.py
Введите целое число: 0
```

Рисунок 2. Результат работы программы из задачи 1

6. Решил следующую задачу: в основной ветке программы вызывается функция `cylinder()`, которая вычисляет площадь цилиндра. В теле `cylinder()` определена функция `circle()`, вычисляющая площадь круга по формуле πr^2 . В теле `cylinder()` у пользователя спрашивается, хочет ли он получить только площадь боковой поверхности цилиндра, которая вычисляется по формуле $2\pi rh$ или полную площадь цилиндра. В последнем случае к площади боковой поверхности цилиндра должен добавляться удвоенный результат вычислений функции `circle()`.

```
C:\Users\slime> slime > LAB11 > progr > zadacha2.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import math
5
6
7  def cylinder():
8      """
9      Вычисляет либо полную площадь цилиндра либо площадь боковой поверхности.
10     """
11
12     def circle(r):
13         """
14         Вычисляет площадь круга.
15         """
16         return math.pi * r ** 2
17
18
19     r = float(input("Введите радиус цилиндра: "))
20     h = float(input("Введите высоту цилиндра: "))
21
22     side_area = 2 * math.pi * r * h
23
24     sw = input(
25         "Если вы хотите найти полную площадь цилиндра - введите 1\n" +
26         "Площадь боковой поверхности цилиндра - введите любой другой символ\n"
27     )
28
29     if sw == "1":
30         base_area = circle(r)
31         full_area = side_area + 2 * base_area
32         print(f"Полная площадь цилиндра: {full_area:.2f}")
33     else:
34         print(f"Площадь боковой поверхности цилиндра: {side_area:.2f}")
35
36
37 if __name__ == '__main__':
38     cylinder()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slme/LAB11/progr/zadacha2.py
Введите радиус цилиндра: 4
Введите высоту цилиндра: 15
Если вы хотите найти полную площадь цилиндра - введите 1
Площадь боковой поверхности цилиндра - введите любой другой символ
1
Полная площадь цилиндра: 477.52
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slme/LAB11/progr/zadacha2.py
Введите радиус цилиндра: 4
Введите высоту цилиндра: 15
Если вы хотите найти полную площадь цилиндра - введите 1
Площадь боковой поверхности цилиндра - введите любой другой символ
a
Площадь боковой поверхности цилиндра: 376.99
PS C:\Users\slime>
```

Рисунок 3. Результат работы программы из задачи 2

7. Решил следующую задачу: напишите функцию, которая считывает с клавиатуры числа и перемножает их до тех пор, пока не будет введен 0. Функция должна возвращать полученное произведение. Вызовите функцию и выведите на экран результат ее работы.

```
C: > Users > slime > LAB11 > progr > zadacha3.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def multiply():
5      """
6      Перемножает введенные числа, до введения 0
7      """
8      result = 1
9      while True:
10         number = float(input("Введите число (введите 0 для завершения): "))
11         if number == 0:
12             break
13         result *= number
14     return result
15
16
17 if __name__ == '__main__':
18     product = multiply()
19     print(f"Произведение введенных чисел: {product}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

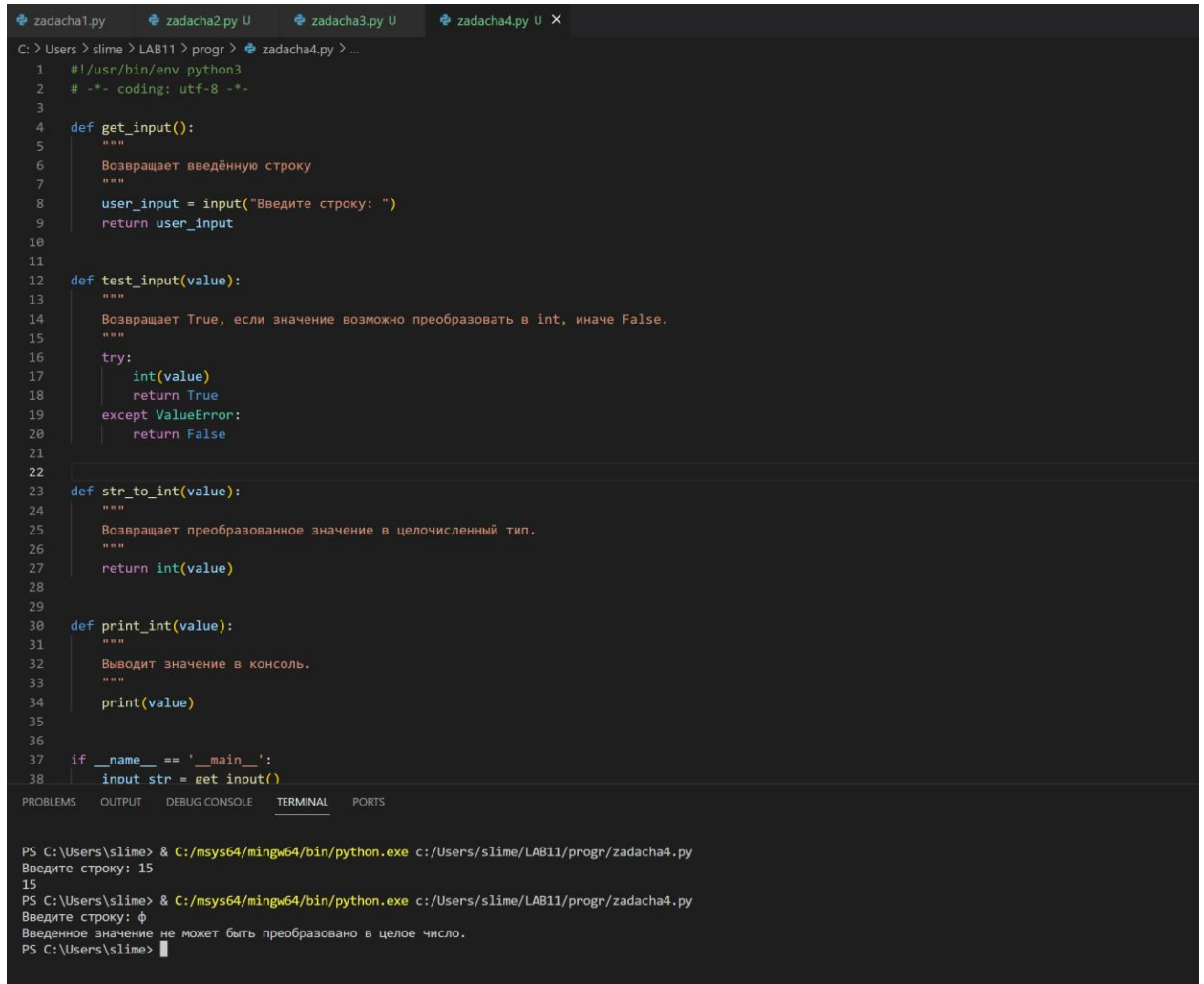
```
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slme/LAB11/progr/zadacha3.py
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slme/LAB11/progr/zadacha3.py
Введите число (введите 0 для завершения): 5
Введите число (введите 0 для завершения): 5
Введите число (введите 0 для завершения): 5
Введите число (введите 0 для завершения): 0
Произведение введенных чисел: 125.0
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slme/LAB11/progr/zadacha3.py
Введите число (введите 0 для завершения): 1.1
Введите число (введите 0 для завершения): 3.3
Введите число (введите 0 для завершения): 51
Введите число (введите 0 для завершения): 0
Произведение введенных чисел: 185.13
PS C:\Users\slime>
```

Рисунок 4. Результат работы программы из задачи 3

8. Решил следующую задачу: напишите программу, в которой определены следующие четыре функции:

- 1) Функция `get_input()` не имеет параметров, запрашивает ввод с клавиатуры и возвращает в основную программу полученную строку;
- 2) Функция `test_input()` имеет один параметр. В теле она проверяет, можно ли переданное ей значение преобразовать к целому числу. Если можно, возвращает логическое `True`. Если нельзя – `False`;
- 3) Функция `str_to_int()` имеет один параметр. В теле преобразовывает переданное значение к целочисленному типу. Возвращает полученное число;
- 4) Функция `print_int()` имеет один параметр. Она выводит переданное значение на экран и ничего не возвращает.

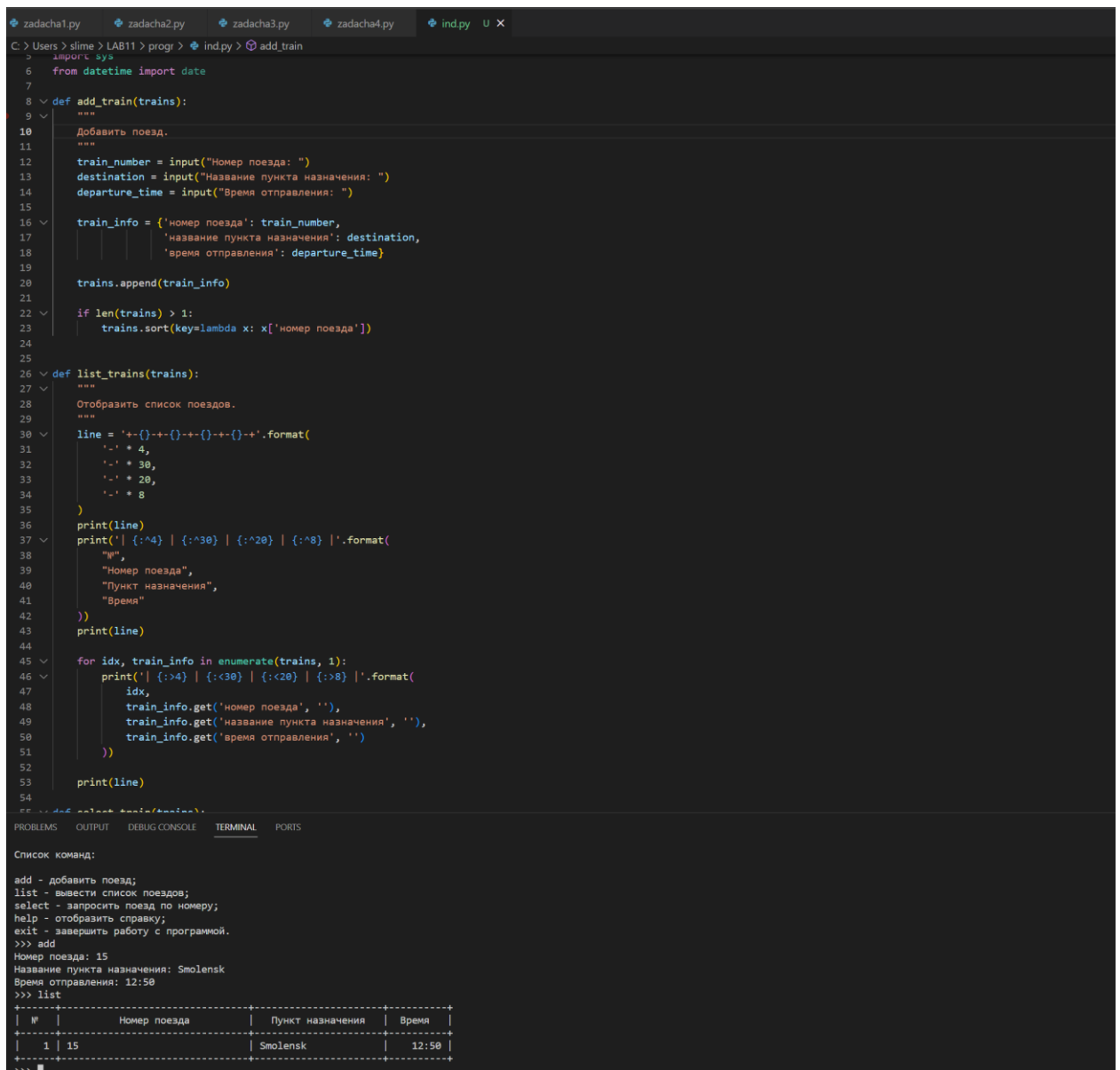
В основной ветке программы вызовите первую функцию. То, что она вернула, передайте во вторую функцию. Если вторая функция вернула True, то те же данные (из первой функции) передайте в третью функцию, а возвращенное третьей функцией значение – в четвертую.



```
zadacha1.py  zadacha2.py U  zadacha3.py U  zadacha4.py U X
C: > Users > slime > LAB11 > progr > zadacha4.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def get_input():
5      """
6      Возвращает введенную строку
7      """
8      user_input = input("Введите строку: ")
9      return user_input
10
11
12 def test_input(value):
13     """
14     Возвращает True, если значение возможно преобразовать в int, иначе False.
15     """
16     try:
17         int(value)
18         return True
19     except ValueError:
20         return False
21
22
23 def str_to_int(value):
24     """
25     Возвращает преобразованное значение в целочисленный тип.
26     """
27     return int(value)
28
29
30 def print_int(value):
31     """
32     Выводит значение в консоль.
33     """
34     print(value)
35
36
37 if __name__ == '__main__':
38     inout_str = get_input()
39
40     if test_input(inout_str):
41         print_int(str_to_int(inout_str))
42     else:
43         print("Введенное значение не может быть преобразовано в целое число.")
44
45 if __name__ == '__main__':
46     inout_str = get_input()
47     if test_input(inout_str):
48         print_int(str_to_int(inout_str))
49     else:
50         print("Введенное значение не может быть преобразовано в целое число.")
51
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slime/LAB11/progr/zadacha4.py
Введите строку: 15
15
PS C:\Users\slime> & C:/msys64/mingw64/bin/python.exe c:/Users/slime/LAB11/progr/zadacha4.py
Введите строку: ф
Введенное значение не может быть преобразовано в целое число.
PS C:\Users\slime>
```

Рисунок 5. Результат работы программы из задачи 4

7. Выполнил индивидуальное задание. Решить индивидуальное задание лабораторной работы 2.6, оформив каждую команду в виде отдельной функции.



```
C:\Users> cd C:\Users\LAB11\progr > ind.py > add_train
> import sys
6 from datetime import date
7
8 def add_train(trains):
9     """
10     Добавить поезд.
11     """
12     train_number = input("Номер поезда: ")
13     destination = input("Название пункта назначения: ")
14     departure_time = input("Время отправления: ")
15
16     train_info = {'номер поезда': train_number,
17                  'название пункта назначения': destination,
18                  'время отправления': departure_time}
19
20     trains.append(train_info)
21
22     if len(trains) > 1:
23         trains.sort(key=lambda x: x['номер поезда'])
24
25
26 def list_trains(trains):
27     """
28     Отобразить список поездов.
29     """
30     line = '+-{}-+-{}-+-{}-+-{}-+'.format(
31         '-' * 4,
32         '-' * 30,
33         '-' * 20,
34         '-' * 8
35     )
36     print(line)
37     print('| {:^4} | {:^30} | {:^20} | {:^8} |'.format(
38         "",
39         "Номер поезда",
40         "Пункт назначения",
41         "Время"
42     ))
43     print(line)
44
45     for idx, train_info in enumerate(trains, 1):
46         print('| {:>4} | {:<30} | {:<20} | {:>8} |'.format(
47             idx,
48             train_info.get('номер поезда', ''),
49             train_info.get('название пункта назначения', ''),
50             train_info.get('время отправления', '')
51         ))
52
53     print(line)
54
55 def select_train(trains):
56     """
57     Выбрать поезд по номеру.
58     """
59     if not trains:
60         print("Список поездов пуст")
61         return
62
63     idx = 1
64     for train_info in trains:
65         print(idx, train_info.get('номер поезда', ''),
66               train_info.get('название пункта назначения', ''),
67               train_info.get('время отправления', ''))
68         idx += 1
69
70     print(line)
71
72 if __name__ == '__main__':
73     trains = []
74     while True:
75         command = input("add - добавить поезд; list - вывести список поездов; select - запросить поезд по номеру; help - отобразить справку; exit - завершить работу с программой. ")
76         if command == 'add':
77             add_train(trains)
78         elif command == 'list':
79             list_trains(trains)
80         elif command == 'select':
81             select_train(trains)
82         elif command == 'help':
83             print("add - добавить поезд; list - вывести список поездов; select - запросить поезд по номеру; help - отобразить справку; exit - завершить работу с программой.")
84         elif command == 'exit':
85             break
86         else:
87             print("Неизвестная команда")
88
89     print("Программа завершена")
```

Список команд:

```
add - добавить поезд;
list - вывести список поездов;
select - запросить поезд по номеру;
help - отобразить справку;
exit - завершить работу с программой.
>>> add
Номер поезда: 15
Название пункта назначения: Smolensk
Время отправления: 12:50
>>> list
```

№	Номер поезда	Пункт назначения	Время
1	15	Smolensk	12:50

Рисунок 6. Результат работы программы из индивидуального задания

Контрольные вопросы

1. Каково назначение функций в языке программирования Python?

Функции в языке программирования Python используются для группировки кода, чтобы он мог быть многократно использован, делая программу более читаемой и легко управляемой. Функции также позволяют разделить большие программы на более мелкие, что упрощает разработку и управление кодом.

2. Каково назначение операторов def и return ?

Оператор def используется для определения функции в Python. Он указывает интерпретатору, что следующий блок кода является телом

функции. Оператор `return` используется для возврата значения из функции.

Когда интерпретатор Python достигает оператора `return`, он возвращает указанное значение и завершает выполнение функции.

3. Каково назначение локальных и глобальных переменных при написании функций в Python?

Локальные переменные объявляются внутри функции и доступны только внутри этой функции. Глобальные переменные объявляются вне функций и доступны во всем коде программы. При написании функций

в

Python, локальные переменные используются для временного хранения данных, в то время как глобальные переменные могут быть использованы в

различных частях программы.

4. Как вернуть несколько значений из функции Python?

Для возврата нескольких значений из функции в Python используется механизм кортежей. Функция может вернуть кортеж, содержащий несколько

значений, и затем эти значения могут быть присвоены различным переменным при вызове функции.

5. Какие существуют способы передачи значений в функцию?

Значения могут быть переданы в функцию в Python через позиционные аргументы, ключевые аргументы и аргументы по умолчанию.

Позиционные

аргументы передаются в порядке, в котором они определены в сигнатуре

функции. Ключевые аргументы передаются с указанием имени параметра.

Аргументы по умолчанию имеют значения по умолчанию и могут быть пропущены при вызове функции.

6. Как задать значение аргументов функции по умолчанию?

Для задания значения аргументов функции по умолчанию в Python используется синтаксис "переменная=значение" в сигнатуре функции. Если значение не передается при вызове функции, будет использовано значение по умолчанию.

7. Каково назначение lambda-выражений в языке Python?

Lambda-выражения в Python представляют собой анонимные функции, которые могут содержать только одно выражение. Они обычно используются в ситуациях, когда требуется небольшая функция в одном месте кода.

8. Как осуществляется документирование кода согласно PEP257?

Документирование кода в Python согласно PEP257 включает в себя использование строк документации (docstrings) для описания модулей, функций, классов и методов. Строки документации должны быть заключены в тройные кавычки и предоставлять информацию о назначении, использовании и возвращаемых значениях функций, классов и методов.

9. В чем особенность однострочных и многострочных форм строк документации?

Однострочные строки документации начинаются и заканчиваются тройными кавычками и предназначены для краткого описания модулей, функций, классов или методов. Они обычно используются для кратких пояснений и описаний.

Многострочные строки документации также начинаются и заканчиваются тройными кавычками, но могут занимать несколько строк и предоставлять более подробное описание. Они обычно используются для более полного и подробного документирования, включая информацию о назначении, использовании и возвращаемых значениях функций, классов и методов.

Вывод: в ходе выполнения работы были приобретены навыки по работе с функциями при написании программ с помощью языка программирования Python версии 3.x